HUMAN-COMPUTER INTERACTION, 2004, Volume 19, pp. 117–149

Beyond Usability Evaluation: Analysis of Human–Robot Interaction at a Major Robotics Competition

Holly A. Yanco
University of Massachusetts Lowell

Jill L. Drury
The MITRE Corporation

Jean Scholtz
National Institute of Standards and Technology

Holly Yanco is a roboticist with an interest in navigation in unstructured environments, assistive technology, and methods for improving shared human–robot control; she is an Assistant Professor of Computer Science at the University of Massachusetts Lowell. **Jill Drury** is a usability engineer and researcher with an interest in evaluating collaborative computing systems; she is an Associate Department Head in the Information Technology Center of The MITRE Corporation and an Adjunct Assistant Professor in the Computer Science Department of the University of Massachusetts Lowell. **Jean Scholtz** is a computer scientist with an interest in evaluation of interactive systems and human–robot interaction; she is a research scientist in the Information Access Division of the Information Technology Laboratory at the National Institute of Standards and Technology.

CONTENTS

ABSTRACT

Human–robot interaction (HRI) is a relatively new field of study. To date, most of the effort in robotics has been spent in developing hardware and software that expands the range of robot functionality and autonomy. In contrast, little effort has been spent so far to ensure that the robotic displays and interaction controls are intuitive for humans. This study applied robotics, human–computer interaction (HCI), and computer-supported cooperative work (CSCW) expertise to gain experience with HCI/CSCW evaluation techniques in the robotics domain. As a case study for this article, we analyzed four different robot systems that competed in the 2002 American Association for Artificial Intelligence Robot Rescue Competition. These systems completed urban search and rescue tasks in a controlled environment with predetermined scoring rules that provided objective measures of success. This study analyzed pre-evaluation questionnaires; videotapes of the robots, interfaces, and operators; maps of the robots' paths through the competition arena; post-evaluation debriefings; and critical incidents (e.g., when the robots damaged the test arena). As a result, this study developed guidelines for developing interfaces for HRI.

1. INTRODUCTION

When there is a disaster, such as an earthquake or terrorist attack, trained professionals search for victims. Often, these professionals make use of rescue dogs; more recently, they have begun to use robots (e.g., Casper, 2002). Robots will play an even greater role in search and rescue missions in the future because they can squeeze into spaces too small for people to enter and can be sent into areas too structurally unstable or contaminated for safe navigation by human or animal searchers.

Robots have been designed for many situations, including museum guides (Thrun et al., 2000) and conference presenters (Simmons et al., 2003). Urban search and rescue, however, is a prime example of a class of safety-critical situations: situations in which a run-time error or failure could result in death, injury, loss of property, or environmental harm (Leveson, 1986). Safety-critical situations, which are usually also time critical, provide one of the bigger challenges for robot designers due to the vital importance that robots perform exactly as intended and support humans in efficient and error-free operations.

Disasters that can serve as field settings for evaluating robot (and human–robot) performance are rare and unpredictable. Therefore, every year, roboticists hold urban search and rescue competitions to speed the development of research, to learn from one another, and to forge connections to the search and rescue community. The research we describe in this article used one of these competitions to investigate issues in human–robot interaction (HRI). Specifically, we studied HRI at the 2002 American Association for Artificial Intelligence (AAAI) Robot Rescue Competition (also known as AAAI–2002). We focused on the effectiveness of techniques for making human operators aware of pertinent information regarding the robot and its environment.

The study had two parts, centering on the performance of four teams in the AAAI–2002 competition and on the use of two of the teams' systems by a domain expert. The competition provided a unique opportunity to correlate objective performance (e.g, number of victims found, number of penalties assessed, percentage of competition arena area traversed) with user interface (UI) design approaches (e.g., degree of information fusion, presence or absence of a computer-generated map display, etc.). The juxtaposition of the team runs with the domain expert use of interfaces allowed us to compare expert (system developer) versus novice (domain expert) use of the interfaces.

The twin goals of our study were to begin developing a set of HRI design guidelines and, more generally, to gain experience in applying human–computer interaction (HCI) and computer-supported collaborative work (CSCW) techniques to the robotics domain. Although much work has been done in the fields of HCI and CSCW to evaluate the usability of interfaces, little of this work has been applied specifically to robotics.

2. RELATED WORK FOR EVALUATION OF HRI

Before any interface (robotic or otherwise) can be evaluated, it is necessary to understand the users' relevant skills and mental models and to develop evaluation criteria with those users in mind. Evaluations based on empirically validated sets of heuristics (Nielsen, 1994) have been used on desktop UIs and Web-based applications. However, current human–robot interfaces differ widely depending on platforms and sensors, and existing guidelines are not adequate to support heuristic evaluations.

Messina, Meystel, and Reeker (2001) proposed some criteria in the intelligent systems literature, but they are qualitative criteria that apply to the performance of the robot only, as opposed to the robots and the users acting as a cooperating system. An example criterion is, "The system … ought to have the capability to interpret incomplete commands, understand higher level, more abstract commands, and to supplement the given command with additional information that helps to generate more specific plans internally" (p. 1).

In contrast, Scholtz (2002) proposed six evaluation guidelines that can be used as high-level evaluation criteria:

1. Is the necessary information present for the human to be able to determine that an intervention is needed?
2. Is the information presented in an appropriate form?
3. Is the interaction language efficient for both the human and the intelligent system?
4. Are interactions handled efficiently and effectively—both from the user and the system perspective?
5. Does the interaction architecture scale to multiple platforms and interactions?
6. Does the interaction architecture support evolution of platforms?

Usability evaluations use effectiveness, efficiency, and user satisfaction as metrics for evaluation of UIs. Effectiveness metrics evaluate the performance of tasks through the UI. In HRI, the operators' tasks are to monitor the behavior of robots (if the system has some level of autonomy); to intervene when necessary; and to control navigation either by assigning waypoints, issuing a command such as "back-up," or teleoperating the robot if necessary. In addition, in search and rescue, operators have the task of identifying victims and their location.

Not only must the necessary information be present, it must also be presented in such a way as to maximize its utility. Information can be present but in separated areas of the interface, requiring users to manipulate windows to gain an overall picture of system state. Such manipulation takes time and can result in an event not being noticed for some time. Information fusion is another aspect of presentation. Time delays and errors occur when users need to fuse a number of different pieces of information.

As robots become more useful in various applications, we think in terms of using multiple robots. Therefore, the UIs and the interaction architectures must scale to support operators controlling more than one robot.

Robot platforms have made amazing progress in the last decade and will continue to progress. Rather than continually developing new user interaction schemes, is it possible to design interaction architectures and UIs to support hardware evolution? Can new sensors, new types of mobility, and additional levels of autonomy be easily incorporated into an existing UI?

We use Scholtz's (2002) guidelines as an organizing theme for our analysis, operationalizing and tailoring them to be specific to the urban search and rescue environment.

Evaluation methods from the HCI and CSCW worlds can be adapted for use in HRI as long as they take into account the complex, dynamic, and autonomous nature of robots. The HCI community often speaks of three major classes of evaluation methods: inspection methods (evaluation by UI experts), empirical methods (evaluation involving users), and formal methods (evaluation focusing on analytical approaches). Robot competitions lend themselves to empirical evaluation because they involve users performing typical tasks in as realistic an environment as possible (for a description of some robot competitions, see Yanco, 2001). Unfortunately (from the viewpoint of performing the empirical technique known as formal usability testing), robot competitions normally involve the robot developers, not the intended users of the robots, operating the robots during the competition. The performance attained by robot developers, however, can be construed as an "upper bound" for the performance of more typical users. Specifically, if the robot developers have difficulty using aspects of the interface, then typical users will likely experience even more difficulty. In addition, robot competitions afford an interesting opportunity (one not attained so far in formal usability testing of HRI) to correlate HRI performance under controlled conditions to HRI design approaches.

Although the AAAI Robot Competition provided us with an opportunity to observe users performing search and rescue tasks, there were two limitations. First, we were not able to converse with the operators due to the time constraints they were under, which eliminated the possibility of conducting think-aloud (Ericsson & Simon, 1980) or talk-aloud (Ericsson & Simon, 1993) protocols, and also eliminated our ability to have operators perform tasks other than those implied by the competition (i.e., search for victims). Second, the competition simulated a rescue environment. Many of the hazards (beyond those incorporated in the arena) and stress-inducing aspects of an actual search and rescue environment were missing. Nonetheless, this environment was probably the closest we could use in studying search and rescue tasks due to safety and time constraints in actual search and rescue missions.

Two patterns were observed in previous HRI empirical testing efforts that limit the insights obtained to date. The first, as mentioned previously, is a tendency for robot performance to be evaluated using atypical users. For example, Yanco (2000) used a version of a usability test as part of an evaluation of a robotic wheelchair system but did not involve the intended users operating the wheelchair (the wheelchair was observed operating with able-bodied occupants). We have started to break this pattern by also analyzing the use of two urban search and rescue robot systems by a fire chief, a more typical user, after the competition runs were completed.

The second pattern that limits HRI empirical testing effectiveness is the tendency to conduct such tests very informally. For example, Draper, Pin, Rowe, and Jansen (1999) tested the Next Generation Munitions Handler/Advanced Technology Demonstrator, which involves a robot that re-arms military tactical fighters. Although experienced munitions loaders were used as test participants, testing sessions were actually hybrid testing and training sessions, and test parameters were not held constant during the course of the experiment. Data analysis was primarily confined to noting test participants' comments such as, “I liked it when I got used to it.” Our study took advantage of the structure inherent in the conduct of the AAAI Robot Competition to keep constant variables such as environment, tasks, and time allowed to complete tasks. In addition, the competition is held annually, which will allow us to track HRI progress and problems over time.

3. METHOD

The two portions of the study consisted of evaluating the interfaces as their developers competed and when the domain expert performed four tasks with each of the interfaces. This section begins with descriptions of the criteria we used for evaluating the interfaces and the evaluation environment we used for both portions of the study. It continues with the methodology used for assessing the interfaces as they were used during the competition. We correlated competition performance with various features in the interface design; therefore, we describe the competition scoring methodology in the fourth subsection. The methodology we used for the domain expert runs comprises the fifth subsection. Finally, we coded the resulting videotapes of both portions of the study using the same coding scheme, which we describe at the end of this section.

3.1. Method for Assessing Interaction Design

An accepted evaluation methodology in HCI is to take a general set of principles and tailor them for use in evaluating a specific application (e.g., see Nielsen, 1993). We operationalized and tailored Scholtz’s (2002) evaluation guidelines as follows to be more specific to the case of HRI in an urban search and rescue context.

“Is the necessary information present for the human to be able to determine that an intervention is needed?” becomes “Is sufficient status and robot location information available so that the operator knows the robot is operating correctly and avoiding obstacles?” “Necessary information” is very broad. In the case of urban search and rescue robots, operators need information regarding the robot’s health, especially if it is not operating correctly. Another critical piece of information operators need is the robot’s location relative to obstacles, regardless of whether the robot is operating in an autonomous or teleoperated mode. In either case, if the robot is not operating correctly or is about to collide with an obstacle, the operator will need to take corrective action.

“Is the information presented in an appropriate form?,” becomes “Is the information coming from the robots presented in a manner that minimizes operator memory load, including the amount of information fusion that needs to be performed in the operators' heads?" Robotic systems are very complex. If pieces of information that are normally considered in tandem (e.g., video images and laser ranging sensor information) are presented in different parts of the interface, the operator will need to switch his attention back and forth, remembering what he or she saw in a previous window to fuse the information mentally. Operators can be assisted by information presentation that minimizes memory load and maximizes information fusion.

"Is the interaction language efficient for both the human and the intelligent system? Are interactions handled efficiently and effectively—both from the user and the system perspective?" Combining these two, they become, "Are the means of interaction provided by the interface efficient and effective for the human and the robot (e.g., are shortcuts provided for the human)?" We consider these two guidelines together because there is little language per se in these interfaces; rather, the more important question is whether the interactions minimize the operator's workload and result in the intended effects.

We are looking at interaction in a local sense, that is, we are focused on interactions between an operator and one or more robots. The competitions currently emphasize this type of interaction but do not provide an environment to study the operator–robot interaction within a larger search and rescue team.

Interactions differ depending on autonomous capabilities of the robots. From the user perspective, we are interested in finding the most efficient means of communicating with robots at all levels of autonomy. For example, if a robot is capable of autonomous movement between waypoints, then how does the operator specify these points? The interaction language must also be efficient from the robot point of view. Can the input from the user be quickly and unambiguously parsed? If the operator inputs waypoints by pointing on a map, what is the granularity? If the user types robot commands, is the syntax of the commands easily understood? Are error dialogues needed in the case of missing or erroneous parameters?

"Does the interaction architecture scale to multiple platforms and interactions?" becomes "Does the interface support the operator directing the actions of more than one robot simultaneously?" A goal in the robotics community is for a single operator to be able to direct the activities of more than one robot at a time. Multiple robots can allow more area to be covered, can allow for different types of sensing and mobility, or can allow for the team to continue operating after an individual robot has failed. Obviously, if multiple robots are to be used, the interface needs to enable the operator to switch his or her attention among robots successfully.

"Does the interaction architecture support evolution of platforms?" becomes "Will the interface design allow for adding more sensors and more autonomy?" A robotic system that currently includes a small number of sensors is likely to add more sensors as they become available. In addition, robots will become more autonomous, and the interaction architecture will need to support this type of interaction. If the interaction architecture has not been designed with these possibilities in mind, it may not support growth.

3.2. Assessment Environment

The robots competed in the Reference Test Arenas for Autonomous Mobile Robots developed by the National Institute of Standards and Technology (NIST; Jacoff, Messina, & Evans, 2000, 2001). The arena consists of three sections that vary in difficulty. The yellow section, the easiest to traverse, is similar to an office environment containing light debris (fallen blinds, overturned table and chairs). The orange section is more difficult to traverse due to the variable floorings, a second story accessible by stairs or a ramp, and holes in the second story flooring. The red section, the most difficult section, is an unstructured environment containing a simulated pancake building collapse, piles of debris, unstable platforms to simulate a secondary collapse, and other hazardous junk such as rebar and wire cages. Figure 1 shows one possible floor plan for the NIST arena. The walls of the arena are easily modified to create new internal floor layouts, which prevent operators from having prior knowledge of the arena map.

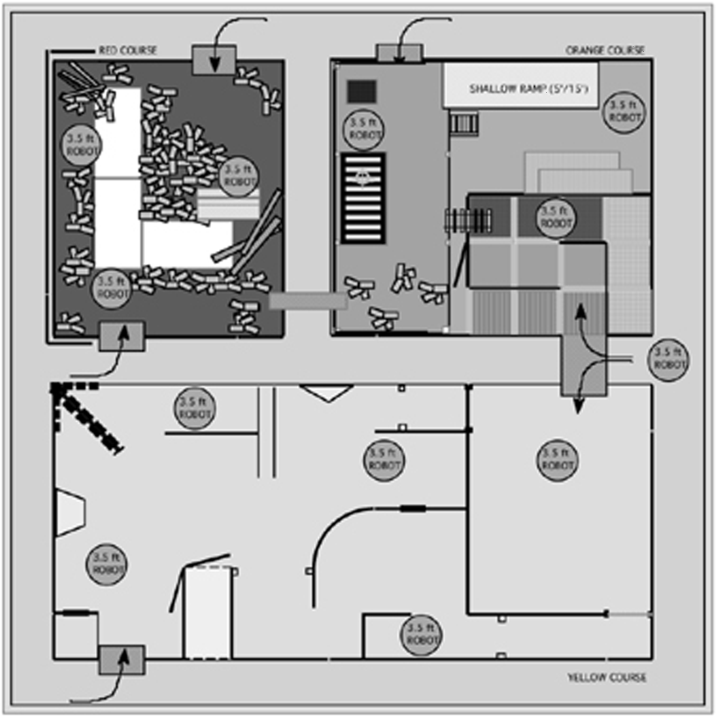
In the arena, victims are simulated using mannequins. Some of the mannequins are equipped with heating pads to show body warmth, motors to create movement in the fingers and arms, tape recorders to play recordings of people calling for help, or all three. Victims in the yellow arena are easier to locate than victims in the orange and red arenas. In the yellow arena, most victims are located in the open. In the orange arena, victims are usually hidden behind obstacles or on the second level of the arena. In the red arena, victims are in the pancake layers of the simulated collapse. Between rounds, the victim locations are changed to prevent knowledge gained during earlier rounds from providing an easier search in later rounds.

Operator stations were placed away from the arena and set up so that the operators would have their backs to the arena. Therefore, the operators were not able to see the progress of their robots in the arena; they had to assess the robots' situations using their UIs.

3.3. Method for Studying Team Performance

Teams voluntarily registered for the competition. We asked them to participate in our study but made it clear that study participation was not a requirement for competition participation. The incentive to participate in the study was the chance to have their robot system used by a domain expert in the second part of the study.

Figure 1. **The National Institute of Standards and Technology test arena for urban search and rescue used by the robots in the competition.**



Participating teams were asked to fill out a questionnaire before the start of the competition. The questions inquired about the robot hardware being used, the type of data provided to the human operator, the level of autonomy achieved by the robot, the maturity of the robot design, and whether the interface was based on a custom (bespoke) or commercial product.

Once the competition began, we observed the operator of each team's robots during the three 15-min runs of the competition. The operator and the interface screen were videotaped. The robots were also videotaped in the arena; cameras were placed in various locations around the arena in an attempt to keep the robot constantly within sight.

We were silent observers, not asking the operators to do anything differently during the competition than they would have already done; our study could not impact on the competition outcome. We could not ask the participants to use the "thinking aloud" protocol, although one participant who was eager to obtain feedback on his interface voluntarily voiced his thoughts as he worked with his robot during the competition. At the conclusion of each run, our observer performed a quick debriefing of the operator via a short postrun interview to obtain the operator's assessment of the robot's performance.

In addition, we were given the scoring materials from the competition judges that indicated where victims were found and penalties that were assessed. We also created maps by hand that showed the approximate paths that the robots took and marked the critical incidents such as hitting objects or victims that occurred during the runs.

3.4. Method for Scoring Team Performance

The scoring algorithm utilized the number of victims found and the accuracy of reporting the location of the victims. The scoring scheme penalized teams for allowing robots to bump into obstacles or victims.[1] The judges recorded a minor victim penalty for bumping into a victim (subtracting .25 from the number of victims found), and a major victim penalty was scored for an event such as causing a pancake layer to collapse on a victim (subtracting 1). Minor damage to the environment, such as moving a wall a small amount, was marked as a minor environment penalty (subtracting .25), whereas major environment damage, such as causing a secondary collapse, was considered a major penalty (subtracting .75).

The scoring formula is:

$$\textit{Performance Score} = (V - P) * A$$

where V = number of victims found; P = penalties; and A = accuracy = 1 if map produced by system, 0.6 if good quality hand-drawn map produced, 0.4 if poor hand-drawn map produced (the accuracy score was determined by the competition judges).

3.5. Method for Studying Domain Expert Performance

After the competition, we had access to a search and rescue domain expert: a special operations fire chief who had participated in training sessions with robots for search and rescue. The goals of the evaluation were to assess ease of learning as well as ease of use. To evaluate ease of learning, the domain expert was asked to explore the interface for 5 min to determine what information was available in the interface. Then the domain expert was given about 5 min

[1] The scoring algorithm used for comparing teams in this study differs from the official scoring algorithm used in the competition (AAAI/RoboCup, 2002). We factored out measures that were unrelated to the interface, such as a measure for calculating a bonus when unique victims were found by different robots.

of training, which would be a realistic amount of training in the field in an emergency condition if the primary (more thoroughly trained) operator suddenly became unavailable (R. Murphy, personal communication, August 2002). After the training, the domain expert was asked to describe the information available in the interface that he did not see during the initial exploration period. Finally, the domain expert was asked to navigate the robot through the arena.

The domain expert was able to verbalize his thoughts as he navigated the robots. He produced a combination of think-aloud and talk-aloud protocols. In general, as he was navigating through the arena, he used the talk-aloud protocol. However, there were a number of times when we experienced technical difficulties, and the chief had to wait for a resolution to the problem before he could proceed. During these times, his verbalizations were more introspective.

3.6. Method for Coding Team and Domain Expert Sessions

Our data consisted of videotapes, competition scoring sheets, maps of robot paths, questionnaire and debriefing information, and researcher observation notes. The richest sources of information were the videotapes. In most cases, we had videotapes of the robots moving through the arena, the UIs, and videos of the operators themselves.

To make the most of the videotaped information, we developed a coding scheme to capture the number and duration of occurrences of various types of activities observed. Our scheme consists of a two-level hierarchy of codes: Header codes capture the high-level events, and primitive codes capture low-level activities. The following header codes were defined: identifying a victim, robot logistics (e.g., undocking smaller robots from a larger robot), failures (hardware, software, or communications), and navigation and monitoring navigation (directing the robot or observing its autonomous motion). Three primitive codes were defined: monitoring (watching the robot when it is in an autonomous mode), teleoperation ("driving" the robot), and UI manipulation (switching among windows, selecting menu items, working with dialog boxes, typing commands, etc.).

Our coding scheme was inspired by the structure of the Natural Goals, Operators, Methods, and Selection Rules Language (NGOMSL) used to model UI interaction (Kieras, 1988). NGOMSL models consist of a top-down, breadth-first expansion of the user's top-level goals into methods, and the methods contain only primitive operations (operators), branch statements, and calls to other NGOMSL methods. Our top-level header codes can be thought of as NGOMSL goal-oriented methods for identifying a victim, navigation and monitoring, or handling robot logistics or failures. Our primitives are not always true primitives (e.g., an activity such as teleoperation can usually be broken down into finer grained motor control actions). However, they are at the lowest level that it makes sense to analyze, and thus they are analogous to NGOMSL primitives.

The coding was done by two sets of researchers. To obtain intercoder reliability, both sets initially coded the same run and compared results. The kappa computed for agreement was .72 after chance was excluded.[2] We then discussed the disagreements and, based on a better understanding, we coded the remaining runs. We did not formally check interrater reliability on the remaining runs as we found in the initial check that we easily agreed on the coding for the events that were observable, but noted that the timing of those events could only be determined within a few seconds. Unfortunately, we could not see the robot when it was in a covered area or when it was in the small portions of the arena that the cameras did not cover.

4. DESCRIPTIONS OF SYSTEMS STUDIED

Eight teams entered the competition. However, we only investigated the HRI of the four teams who found victims during their runs; these teams were also the top-ranked teams. Teams that were unable to find victims most often had hardware failures and no significant amount of HRI to study. In this section, we describe each of the four systems in our study, including the UI and the robot hardware. A summary of the systems is given in Figure 2.

4.1. Team A

Team A developed a heterogeneous team of five robots—one iRobot® ATRV-Mini and four Sony AIBOs®, for the primary purpose of research in computer vision and multiagent systems.[3] They spent 3 months developing their system for the rescue competition. All robots were teleoperated serially. The AIBOs were mounted on a rack at the back of the ATRV-Mini. The AIBOs needed to be undocked to start their usage and redocked after they were used if the operator wanted to continue to take them with the larger robot.

Team A developed two custom UIs, which were created for use by the developers and were not tested with other users before the competition. There was one UI for the ATRV-Mini and another for the AIBOs. The UIs ran on separate computers. Communication between the UI and the robots was accomplished using a wireless modem (802.11b).

[2] When chance was not factored out, the agreement was .8.

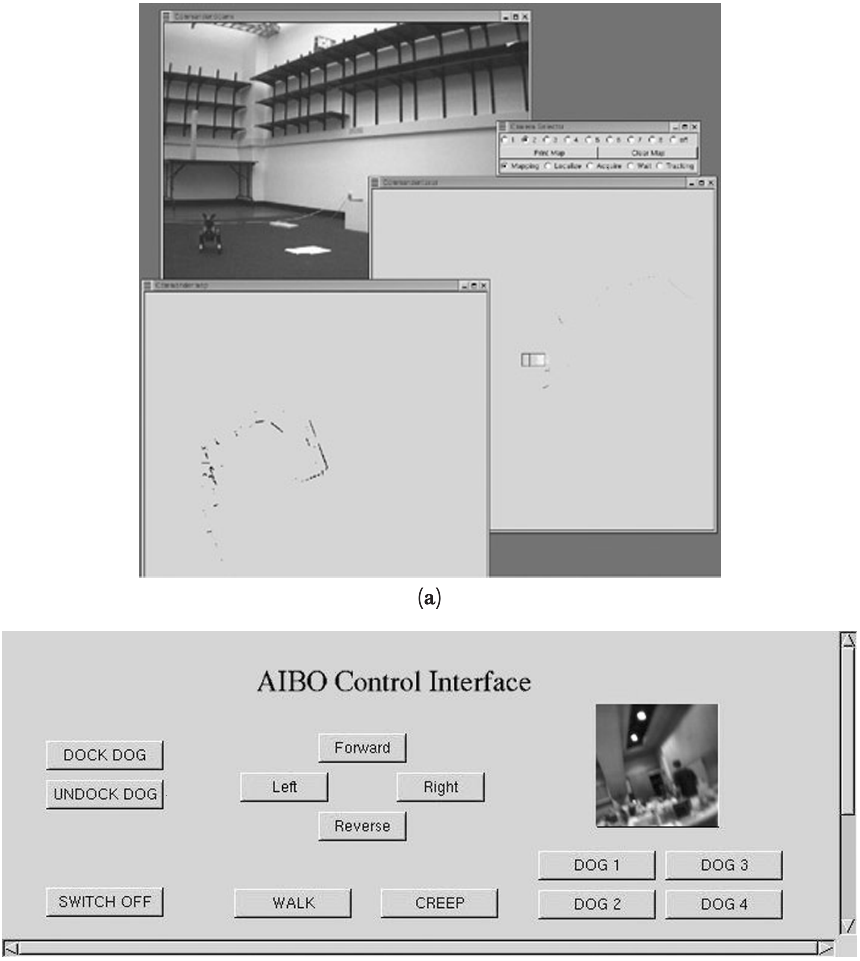
[3] The identification of any commercial product or trade name does not imply endorsement or recommendation.

Figure 2. **Summary of system characteristics.**

Variables	Team A	Team B	Team C	Team D
Platform and autonomy characteristics	One iRobot ATRV-Mini and four Sony AIBOs, teleoperated serially	One iRobot ATRV-Jr. with a range of operator–selectable autonomy levels	Two RWI Magellan Pros, teleoperated or avoid obstacles while moving toward goals	Two custom robots (one wheeled and one tracked), teleoperated serially
Sensors	ATRV-Mini: Video (circle of 8 cameras), current laser scan (raw and processed for map); odometry and laser scan fused for map. AIBOs: Video, no fusion	Video, thermal imaging (raw), infrared, bump, laser scan, and sonar; data from last four sensors fused for sensor map	Video, sonar, infrared; data from last two sensors fused for overhead map (evidence grid)	Both have video and no sensor fusion
Interfaces	ATRV-Mini: Multiple windows for video, map, raw laser scan, camera control; keyboard control. AIBOs: video window; keyboard or GUI control	Touch screen with windows for sensor status, battery/velocity/tilt, video (actually displayed on another monitor), sensor map, environment map; control via joystick and touch screen	GUI: Split screen for two robots with video on top and map on bottom. Text-based interface: 14 text and four graphic windows. Control via keyboard	Two displays used: One for video feed, other for pre-entered map of arena; control via keyboard

Note. GUI = graphical user interface.

Figure 3. **(a) Team A's interface for the iRobot ATRV-Mini. (b) Team A's interface for the Sony AIBOs.**



(a)

(b)

The UI for the ATRV-Mini, shown in Figure 3a, had multiple windows. In the upper left-hand corner was a video image taken by the robot, updated once or twice each second. In the lower left-hand corner was a map constructed by the robot using the SICK® laser scanner and odometry (SICK, Inc., Bloomington, MN). In the lower right-hand corner, the raw laser scan information was presented as lines showing distance from the robot. The upper right-hand corner had a window with eight radio buttons labeled 1 through 8 to allow the user to switch camera views. The operator drove the robot using keys on the keyboard to move forward, backward, right, and left.

The UI for the AIBOs, shown in Figure 3b, had a window with the video image sent from the robot. The operator controlled the robots either using buttons on the graphical UI (GUI) or by using the keyboard. (The domain expert controlled the AIBOs using the keyboard because of a problem with the GUI at that time.)

4.2. Team B

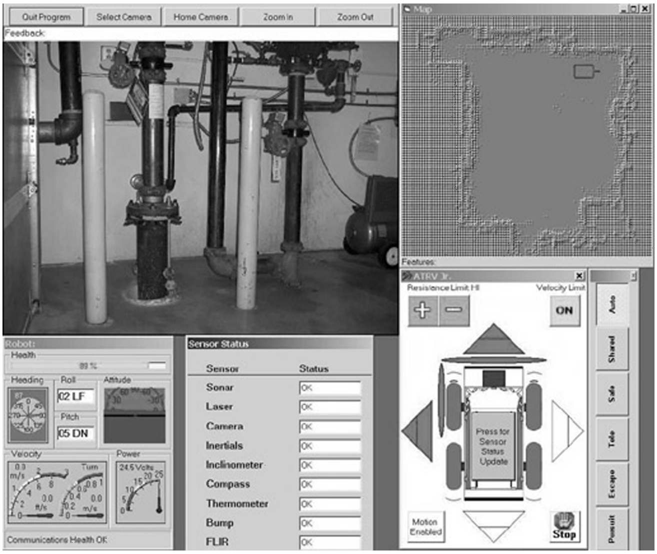
Team B had been developing their robot system for use in hazardous environments for less than 1 year. The robot was an iRobot ATRV-Jr. Communication was achieved through a proprietary, low-bandwidth communication protocol over 900 MHz radio.

The custom UI, shown in Figure 4, was developed for expert users and tested with novice users and experienced operators. The interface was displayed on a touch screen. The upper left-hand corner of the interface contained the video feed from the robot. Tapping the sides of the window moved the camera left, right, up, or down. Tapping the center of the window recentered the camera. (During the competition, the video window had not yet been finished, so the video was displayed on a separate monitor. However, the blank window was still tapped to move the camera.) The robot was equipped with two types of cameras that the operator could switch between: a color video camera and a thermal camera.

The lower left-hand corner contained a window displaying sensor information such as battery level, heading, and tilt of the robot. In the lower right-hand corner, a sensor map was displayed, showing filled red areas to indicate blocked directions. In the picture of the previous interface, a map of the environment can be seen in the upper right-hand corner. Although this space was left for a map during the competition, the software for building and displaying maps had not yet been created, so no maps were provided to the operator.

The robot was controlled through a combination of a joystick and the touch screen. To the right of the sensor map, there were six mode buttons: Auto (autonomous mode), Shared (shared mode, a semi-autonomous mode in which the operator can "guide" the robot in a direction but the robot does the navigation and obstacle avoidance), Safe (safe mode, in which the user controls the navigation of the robot, but the robot uses its sensors to prevent the user from driving into obstacles), Tele (teleoperation mode, in which the human controller is totally responsible for directing the robot), Escape (a mode not used in the competition), and Pursuit (also not used in the competition). Typically, the operator would click on one of the four mode buttons then start to use the joystick to drive the robot. When the operator wished to take a closer look at

Figure 4. **Team B's user interface.**



Health 89 %

Roll 02 LF

Pitch 05 DN

24.5 Volts

Communications Health OK

Sensor Status

Sensor	Status
Sonar	OK
Laser	OK
Camera	OK
Inertials	OK
Inclinometer	OK
Compass	OK
Thermometer	OK
Bump	OK
FLIR	OK

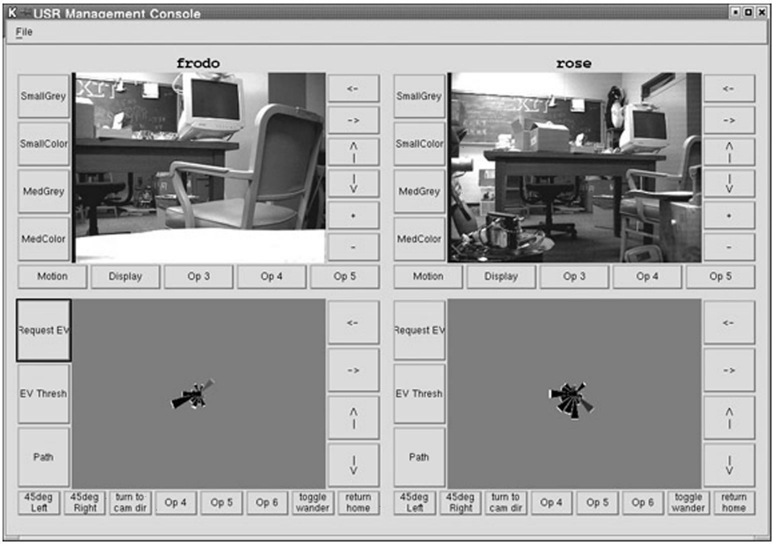
something, perhaps a victim or an obstacle, he would stop driving and click on the video window to pan the camera. For victim identification, the operator would switch over to the thermal camera for verification.

4.3. Team C

Team C had developed their robots for less than 2 years as a research platform for vision algorithms and robot architectures. They used two identical robots, RWI® Magellan Pros. Communication between the UI and robots was achieved with an radio frequency modem.

The robots had a mixed level of autonomy: They could be fully teleoperated, or the robots could provide obstacle avoidance while achieving a specified goal. The robots could run simultaneously, but were operated serially. Waypoints were used to generate maps from the robot's current location to the starting point. The operator commanded the robot by giving it relative coordinates to move toward. The robot then autonomously moved to that location using reactive obstacle avoidance. The robot's ability to carry out a command without assistance allowed for the perception that the operator is moving both robots "at once," although he was controlling them serially. It was the operator's trust in the robots' autonomy that allowed this type of operation; the operator did not need to monitor the progress of one robot while commanding the other.

Figure 5. **Team C's graphical user interface.**



A custom interface, shown in Figure 5, was developed for a "sophisticated user" (according to the developers). Team C started Run 1 using a GUI, but switched to a text-based interface when there were command latency problems with the GUI. In the GUI, the screen was split down the middle; each side was an interface to one of the two robots. The top window for each robot displayed a current video image from the robot, and the bottom window displayed map information.

In the alternative text-based interface, used in the remainder of the runs, the screen had 14 text windows and four graphic windows, one half for each of the robots. Seven text windows were used for the following: the interprocess communication server, the navigation module, the vision module, the mapping module, the navigation command line, a window for starting and monitoring the visual display, and a window for starting and monitoring the map display. Two graphic windows were used for displaying the camera image and the map image. The computer was running an enlarged desktop during the competition, and the operator sometimes needed to switch to another part of the desktop (effectively switching to another display) for other pieces of the interface. The robots were controlled with keystrokes.

4.4. Team D

Unlike the other three systems, Team D developed their robots for search and rescue over the previous year. They had custom-built robots, one wheeled and one tracked, both with the same sensing and operating capabilities. The robots were teleoperated serially. A wireless modem was used to communicate between the UI and the robots.

Team 4 developed a custom UI on two screens.[4] One monitor displayed the video feed from the robot that was currently being operated. The other monitor had a pre-entered map of the arena on which the operator would place marks to represent the locations of victims that were found. The robots were driven with keyboard controls.

5. RESULTS

We present two types of results for the teams: the objective measures from the competition and the results of our coding. We also present the coded results of the domain expert's performance along with his talk-aloud and think-aloud protocols. Finally, we analyze performance using the Scholtz (2002) guidelines from Section 3.1.

5.1. Team Runs

Each team had three 15-min runs during the competition. We only coded Runs 1 and 3 due to the failure of the video data capture equipment during Run 2. The total times are in some cases less or more than the allotted 15 min. It was sometimes difficult to discern the actual starting time for the competition to coordinate the start of data capture, which resulted in shorter times. In addition, in Team A's first run, a tape change caused us to lose some of the data from the run. Longer times resulted from a judge failing to stop the run at exactly 15 min.

Figure 6 shows the percentage of time spent in each of the primary header code activities. The majority of time for most runs was spent navigating, followed by identifying victims. Time spent in logistics or failures was time taken away from looking for victims.

Team scores are shown in Figure 7. Because we did not analyze the HRI in Run 2, we only consider Runs 1 and 3 in the scoring. Using the scoring algorithm in Section 3.4, the rankings for the two rounds would be as follows: 1st place, Team A; 2nd place, Team C; 3rd place, Team D; and 4th place, Team B.[5]

[4] We were unable to obtain a screen shot of this interface from its designers.

[5] The actual rankings in the competition, which included Run 2 as well as other measures such as in which part of the arena victims were found, were as follows: 1st place, Team D; 2nd place, Team C; 3rd place, Team A; and 4th place, Team B.

Figure 6. **Time spent in each of the primary header codes for competition runs.**

			% Time			
Teams	**Run**	**Total Time**	**Navigation–Monitoring Navigation**	**Victim ID**	**Failure**	**Logistics**
A	1	10:39	46	51[a]	0	3
	3	14:45	62	18	19[b]	1[c]
B	1	14:33	81[d]	19	0	0
	3	16:42	77	23	0	0
C	1	13:26	59	23	17[e]	0
	3	14:39	69	12	18[f]	0
D	1	15:12	55	32	0	12[g]
	3	13:30	87	4	0	9

Note. ID = identification.

[a] Includes navigation to get a new angle for victim ID after a judge said that the first image was unclear. [b] Wireless modem failures. [c] In addition, about 25% of the victim ID time was spent in logistics while deploying AIBOs. [d]The operator spent 90% of this navigation time in a confused state. However, the equipment had not malfunctioned, so this was not coded as a failure. [e]Graphical user interface latency, panoramic image failure. [f]Panoramic image failure, vision system on one robot failed midway through run. [g]Switching between two robots.

Figure 7. **Team scores computed using the algorithm in Section 3.4.**

Teams	**Run**	**No. of Victims**	**Penalties**	**Accuracy**	**Score**	**Team Total**
A	1	4	8 × 0.25	1.0	2.0	
	3	3	6 × 0.25	1.0	1.5	3.5
B	1	3	5 × 0.25	0.6	1.05	
	3	0	1 × 0.25 + 3 × 0.75	—	Negative	< 1.05
C	1	3	0	0.4	1.2	
	3	4	3 × 0.25	0.6	1.95	3.15
D	1	6	9 × 0.25	0.5[a]	1.875	
	3	3	4 × 0.25	0.6	1.2	3.075

[a] This number reflects a penalized accuracy score, as determined by the judges. There was some question as to whether advance knowledge of the arena layout had been obtained.

5.2. Domain Expert Runs

Ease of Learning

The domain expert, a special operations fire chief trained in search and rescue and with experience using robots, used two systems: Team A (teleoperated) and Team B (different autonomy modes). We started each session with a short amount of time for the chief to explore the interface without instruction. After this period, we asked him to state what he could figure out about the interface. Then the system developers explained the interface to him, and we asked the chief what was in the interface that he had not seen before.

For Team B, the chief said that there was no real-time video (the team was having trouble with their video link at this time, so they were only sending about one frame per second). He noted that there were sensors around the robot, pointing to the sensor map in the lower right-hand corner, and that the map appeared to be displaying proximity information. After the chief talked with Team B's developer, he stated that he had learned about the control modes for the robot.

For Team A, the chief said that he saw a laser map on the lower right, a video display on the upper right, an ultrasonic map on the left, and a data window under that. He could not see how to drive, but thought he would do it using the arrow and the mouse. After the developer's explanation, the chief learned that the window on the left did not have an ultrasonic map, but was instead displaying a map created as he drove using the laser scan and odometry. He also learned how to control the robot and that there was a ring of cameras on top of the robot for the video window. A window with radio buttons labeled 1 through 8 was used to switch from one camera view to another in the video window.

Ease of Use (Performance)

The chief had been a judge for the competition, so he was more familiar with the arena at the time of his runs than any of the competitors had been. Figure 8 shows the amount of time the domain expert spent in each of the primary header codes. The times shown in the table include the time that the expert was using the systems, not any time that he was speaking to the system developer or the researchers.

We observed the chief relying heavily on the live video for navigation. He would drive, change camera angles, then resume driving. We discuss the primary use of video further in Section 5.3.

5.3. Evaluation Using Tailored Scholtz Guidelines

We use the performance of the teams and of the domain expert, the results of coding activities of the operators during the competition runs, and an examination of critical incidents to discuss Scholtz's (2002) guidelines from Section 3.1.

Figure 8. **Percentages of run time spent in each of the primary header modes for the domain expert's runs.**

		% Time			
Teams	**Total Time**	**Navigation–Monitoring Navigation**	**Victim ID**	**Failure**	**Logistics**
A	18:43	93	0[a]	2[b]	5
B	25:35	97	3	0	0

Note. ID = identification.
[a] Victims were removed from the arena during the chief's runs. [b] Communication failure: video signal was not updating.

Is Sufficient Status and Robot Location Information Available so That the Operator Knows the Robot Is Operating Correctly and Avoiding Obstacles?

The number of penalties for each team is shown in Figure 9. Note that Team A's two different types of robots are listed separately, because the ATRV-Mini has dramatically different sensor capabilities than the AIBOs. Although another team, Team D, also fielded robots of different types, their robots differed only in their navigation properties (one type was tracked and the other was wheeled); otherwise, their sensor suites and operational capabilities were identical.

We had thought that Team B would fare slightly better than they did. Team B's operator experienced serious confusion when he forgot that his robot's video camera was pointing in a direction other than straight ahead. This confusion resulted in more than one half of Team B's first run (8½ min) being wasted. The interface did not provide any reminders that the video camera was pointing off center, so this lack of awareness of robot state (rather than a paucity of sensor data) caused him to run into more obstacles and find fewer victims than he might have otherwise.[6] We are unsure why he also had a poor Run 3. During this run, the operator was frustrated that his robot was "too big" to navigate in the small areas of the arena. In fact, he did have the largest robot in the competition.

We saw several specific instances where operators were unaware of robot locations and surroundings. In several cases (e.g., Team D during Run 1), there was not enough awareness of the area immediately behind the robot, causing the robot to bump obstacles when backing up. Even when

[6] This problem was corrected by the developers before other runs by changing the program to recenter the camera.

***Figure 9.* Number of penalties incurred by the teams.**

Teams	Run	Arena Penalties	Victim Penalties	Rank[a]
A (ATRV-Mini)	1	1 minor[b]	0	
	3	4 minor	0	2
A (AIBOs)	1	7 minor[c]	0	
	3	2 minor	0	—
B	1	3 minor	2 minor	
	3	1 minor, 3 major	0	3
C	1	0	0	
	3	3 minor	0	1
D	1	9 minor	0	
	3	4 minor	0	4

[a] 1 is the best, 4 is the worst bumping record overall based on numbers of bumps. AIBOs are not ranked because they were used for only short periods of time. [b] The ATRV-Mini was used for approximately 12 min during each of Runs 1 and 3. Normalizing to 15 min would result in 1.25 and 5 minor arena penalties, which does not affect Team A's overall ranking. [c] The AIBOs were used for approximately 3 min during Runs 1 and 3.

moving forward, several operators (e.g., Team B during Run 3) hit walls and were not aware of doing so. One of Team A's robots was trapped under fallen Plexiglas®, but the operator was never aware of this situation. Because they did not have precise awareness of the area immediately around the robot, operators (e.g., Team B during Run 3) had a difficult time maneuvering the robots in tight spaces.

One of the debriefing questions we asked after each run was how the operator perceived the performance of the run. Surprisingly, Team B's operator stated after Run 3 that he had not hit anything during the run. However, his perceptions did not correspond with reality; he had incurred one minor and three major arena penalties during this run. Clearly, the operator did not have sufficient awareness of the robot, its surroundings, and its activities.

The chief's bumping performance is shown in Figure 10. Although he was not scored, we marked the times that he hit objects just as was done for the teams. These penalties were marked over the full length of the chief's runs, which were about 10 min longer than an average team run.

While using Team A's system, the chief asked twice if someone was watching in the arena. The first time he said he was not sure if the robot was clear of a wall. The second time he thought the robot might be caught on a cable, but he was told that the robot was clear. To resolve his awareness problems, he deployed an AIBO from the ATRV-Mini and positioned the camera on the ATRV-Mini so that he could view the AIBO while he was teleoperating it.

The chief had begun to experiment with Team B's system earlier and stopped due to wireless interference. He did not feel comfortable relying on

Figure 10. **Penalties incurred by the chief during his runs.**

Teams	Arena Penalties	Victim Penalties
A (ATRV Jr.)	0	0
A (AIBOs)	0	0
B	2 minor, 6 major	0

the sensor display, the single frame video images updated infrequently, and various modes of autonomy for navigation. In this early run, the chief was using the safe mode of navigation and was unable to understand why he could not navigate through a perceived opening. He put the robot in teleoperation and discovered that the "opening" was covered with Plexiglas, but only when people called from the arena area to state that the robot had charged through the panel.

When using both systems, the chief adjusted the camera views frequently, but even then he had difficulty knowing where the robot was. The team operators using these systems relied far less on moving the cameras around to acquire awareness than the chief did. Team B's operator relied on the sensor data and used various modes of autonomy. He used the camera views when he was identifying a victim. However, he also had imperfect awareness; there were a number of instances when he bumped into obstacles and was penalized in the scoring but never noticed this during the run. Team A's operator used the dynamically created map and the laser scanning data for navigating, but he also had suboptimal awareness. When one of the AIBOs fell off the ATRV-Mini, the operator was completely unaware of it.

The developers seemed to feel more comfortable relying on sensor data other than video, which may have provided a false sense of security as their penalty scores reflected their lack of awareness. The chief, a novice user, was more cautious and, although he commented about the usefulness of the sensor data, he still relied heavily on live video feeds that proved to be problematic. Further, not all of the necessary information was presented to users; more information was needed regarding the awareness of the relation of the robot to its environment, as evidenced by a number of bumping incidents.

Is the Information Coming From the Robots Presented in a Manner That Minimizes Operator Memory Load, Including the Amount of Information Fusion That Needs To Be Performed in the Operators' Heads?

Team D had the only system in the competition that had no information fusion in the system, using only video. Team A had the only system that presented a map in the display that included the walls of the arena. This map allowed the operator to see where he had been so that he could hopefully avoid covering the same territory numerous times. Team C also had a map in their interface, but it presented only the sonar readings of the robot as it moved through the arena. No corrections were made for dead reckoning errors. Although Team A's map looked like a floor plan, Team C's map looked like a fat line composed of black triangles. Figure 11 shows that Team A had better coverage than all teams, with the exception of Team D for Run 1.[7] The two teams with maps, A and C, scored above (1st and 2nd, respectively) the two teams without maps, B and D (4th and 3rd, respectively); see Figure 7 for a summary of the scoring.

Although additional sensor information should provide additional awareness as a general rule, this rule does not hold true if more information is provided but the information is not integrated into the displays in a way that an operator can use. In general, lack of data fusion, other than that contained in maps, hindered operators' ability to quickly obtain an understanding of the robot's status and location. For example, for Teams A and B, the video image was presented separately from the sonar or laser ranging sensor data, in opposite corners of the display screen. Such separation requires the operator to mentally synthesize the data as opposed to having the interface provide a combined picture.

Presenting related data in opposite corners of the display is an example of how the displays were not laid out for maximum efficiency or memory load minimization. Evidence of this trend can be seen by the fact that operators spent a large percentage of the time in UI manipulation. Various types of information were, in general, presented in separate windows so that operators spent significant time periods moving between windows. Operators then had to remember what was in one window and combine it with information in other windows. Some operators needed to constantly glance between video and other data, or move between windows on the display, while mentally fusing the various pieces of information.

When using System A, the chief initially noted that he relied primarily on the laser for navigation. However, his primary navigation method was to stop teleoperating the robot and to change the view of the camera to look around. The chief used a similar method to drive Team B's system. He relied heavily on live video and commented when the reception was particularly bad. However, video can miss some types of obstacles, as evidenced by the fact that the chief drove through a Plexiglas panel.

During the first run, the Team B operator moved the robot's video camera off center to look at a victim for identification, and also switched to his thermal

[7] There was some question as to whether Team D had prior knowledge of the arena for Run 1.

Figure 11. **Amount of the arena covered.**

Teams	Run	Coverage
A	1	50% yellow
	3	35% yellow
B	1	20% yellow, 5% orange
	3	35% yellow, 10% orange
C	1	30% yellow
	3	35% yellow
D	1	80% yellow
	3	15% yellow, 10% orange, 5% red

camera to verify that it was a live victim. After the victim identification, the operator switched to shared mode to allow the robot to get out of a tight space with less operator intervention. At this point, the operator forgot that he had turned his camera to the left. When he switched back to safe mode, he found that the results of his actions did not correspond to the video image he saw. This confusion resulted in the operator accidentally driving the robot out of the arena into the crowd and bumping into a wall trying to get back into the arena. The turned camera also resulted in substantial operator confusion (we recorded quotes such as, “it’s really, really hard”; “I got disoriented”; and “oh, no!”). During the third run, Team B’s operator did not have good visibility into the areas behind the robot, making it difficult for him to maneuver it out of narrow spaces (“this is very difficult”). After the third run, Team B’s operator commented that he had not bumped anything, yet four bumping penalties were assessed by the judges.

Team C started a run using a GUI, but within 2 min, the operator determined that there was too much lag time between command issuance and response. As a result, he shut down the GUI and brought up seven windows that formed an earlier version of the interface (the debugging version). It took a little over 1½ min for the operator to shut down the GUI and bring up all the windows for the earlier interface version. In this interface, the operator needed to shuffle through the seven windows to view different types of information and enter commands in several of the windows.

Are the Means of Interaction Provided by the Interface Efficient and Effective for the Human and the Robot (e.g., Are Shortcuts Provided for the Human)?

We saw evidence of inefficient interaction mechanisms that resulted in the user having to switch windows or modes frequently, primarily because the output of each sensor seemed to be provided in a different window. Further, we noted instances where interactions were not effective. The prime example of an ineffective interaction was the case where the operator's efforts to navigate the robot through the arena were unsuccessful due to the fact that he had forgotten that he had previously changed the pointing angle of the video camera from a straight-ahead orientation. Because the interface provided no clues to remind the user that he had forgotten to restore the video camera angle, he persisted in navigating in the wrong directions.

Does the Interface Support the Operator Directing the Actions of More Than One Robot Simultaneously?

The amount of work an operator needed to do to use a robot (via the UIs in the competition) was sufficiently high so that it was unrealistic to expect an operator to control multiple robots simultaneously. Although the systems of Teams A, C, and D were designed to operate with more than one robot simultaneously, in practice the robots were controlled serially. (Recall, however, that Team C was able to have more that one robot navigating at a time due to the autonomy of the systems. However, the operator could only focus on one robot at a time due to the split interface.) Facilitating additional autonomy would help workload, but some amount of monitoring would still be necessary.

With the teams' current UI designs, virtually all of the operators' attention was needed to run one robot at a time due to several reasons. First, as mentioned previously, operators were busy integrating information from the video and the other portions of the interface (e.g., the map showing the current location of the robot in x–y space, thermal images, and video images). Second, there was a high overhead cost to switch from operating one robot to another. All of the windows were duplicated for each robot, rather than having the information integrated into one set of windows. In fact, our coding revealed that Team C, the only team to field two robots simultaneously, spent 7% of their navigation time giving commands to the UI in Run 1, in which one robot was used. During Run 3, when two robots were operated, 13% of the navigation time was spent issuing commands to the robots. Doubling the number of robots doubled the number of commands. Clearly, there will be problems when scaling up, even when robots have some autonomy.

Three of the four competition teams fielded more than one robot. The fact that only one (Team C, Run 3) of the teams operated more than one robot at a time is indicative that their interaction architectures are not appropriately scaled to handle interactions with multiple robots simultaneously. The approach taken to adding multiple robots seems to be to add another set of windows, where many of the windows display only one type of sensor data. With this approach, the user quickly runs out of screen real estate and the cognitive power to mentally fuse the appropriate information for each robot.

Further, each of the interfaces examined makes the user completely responsible for gathering awareness of the robot's state and location by means of moving the video camera around.[8] Hence, we saw many short periods of navigation with lots of gathering awareness in between, where the robot stops moving as the operator manipulates the cameras. Such an approach is difficult to do for more than one robot simultaneously.

Will the Interface Design Allow for Adding More Sensors and More Autonomy?

The interaction architectures we studied do not support robot evolution. Robot evolution usually involves additional sensors and more autonomy; more sensors will require more windows if the current interaction architectures are extended. Although one robot UI we examined does support various modes of autonomy that could ease operator workload, it currently falls to the operator to determine which mode should be used and to switch the robot as necessary. An examination of the percentage of navigation time spent in each of three autonomy modes[9] for Team B, as well as the number of mode switches made during the run, shows that Team B's operator made 20 mode switches in Run 1 and 19 mode switches in Run 2. The chief changed modes 12 times during his run, with the majority of the switches occurring at the end of his time with the robot.

It would be more helpful if the robot could determine the necessary mode based on sensor information and suggest it to the operator, rather than relying on the operator to constantly revisit the decision regarding the optimal mode.

6. DISCUSSION

6.1. Victim Identification

The main purpose of search and rescue robots is to locate victims. A victim must be accurately identified, and an accurate location must be determined so

[8] No other sensors could be manipulated by the interface; if the user wanted to get a different view using nonvideo sensors, the robot would need to be moved.

[9] The operator never used teleoperation, which did not provide any sensor mitigation.

that the rescue teams can construct plans to reach the victims. In Run 3, Team B found no victims, yet spent 23% of the time trying to identify victims that the operator thought he saw. In Run 1, Team A spent additional time obtaining a clearer image of a victim for positive identification.

Sending rescue teams to extract victims is not without risks. Therefore, the operator needs to be reasonably confident of his victim assessment. Video is currently the most utilized means of victim identification, but additional sensors are needed to more accurately assess victim state. Video transmission is difficult even in semicontrolled circumstances such as the competitions. In actual search and rescue situations, the interference in communications will likely be worse. Relying on video alone makes victim identification difficult.

6.2. Time on Tasks

Our analysis showed that failures take up a good percentage of time during runs. Two teams lost time to failures. Failure types differed from communications losses to other issues such as latency. GUIs need to have a low latency time if the operators are using teleoperation to control the robot. Real-time situation awareness is an issue for all types of control, which is hampered by high latencies.

A large percentage of time was also spent in logistics. Multiple robots are beneficial especially if different sized robots are being deployed (e.g., using smaller robots to probe small voids). However, deployment mechanisms need to be carefully analyzed for maximum efficiency. Team A used multiple robots with a low percentage of time devoted to logistics. However, when the chief (a less experienced operator) deployed a second robot, a slightly elevated percentage of time was needed for logistics.

6.3. Navigation in a Difficult Environment

Bumping into walls is penalized in the competitions. In an actual disaster, a robot that bumped a wall could trigger more damage and cause a wall to collapse. The test arena has a number of partitions that simulate walls and windows. Different wall coverings are difficult to detect with various sensors. For example, the results of our study showed the difficulty of relying on vision to detect Plexiglas.

Obstacles in the arena consist of office furnishings and building material debris: chairs, papers, Venetian blinds, pipes, electrical cords, and bricks or cinder blocks. As robot mobility increases, the test arena will incorporate more realistic obstacles. The goal is to avoid these obstacles, but that is not always possible. Robots will become entangled and will need help from the operator to get free.

6.4. Operator Information

The information needs of the operator fall into several categories: information about the status of the robot, information about the robot's environment, and information about victims found in the environment. Information about the status of the robot and the robot's environment is necessary for real-time monitoring and control or supervision of the search. The operator uses information about victim state and location to ensure coverage. In competitions, the accuracy of maps is verified by giving the information to judges; in real situations, people would be sent into a building to rescue reported victims.

In this analysis we have focused on the information needed by the operator to navigate the test arena and to locate victims. We looked at the interactions between the operator and one or more robots. These constraints were determined by the nature of the competition and the capabilities of the teams participating in the search and rescue competition. As capabilities of robots improve we hope to see entries that have robot–robot interactions and operator–operator interactions. The competition limits us to studying the operator–robot pairing rather than allowing us to study the larger context of an entire search and rescue team at a disaster site. For the time being, we can look to studies such as Burke, Murphy, Coovert, and Riddle (2004) for insights into HRI in the larger context of urban search and rescue.

7. CONCLUSIONS AND GUIDELINES

Our study of the operator role in human–robot teams looked at systems ranging from complete teleoperation to systems allowing some degree of autonomy. We looked at systems with sensory input ranging from video only to robots with sensor suites that included laser ranging, sonar, infrared, thermal cameras, and video cameras. We found that more sensor types do not necessarily increase awareness, especially if the sensor data is not well fused into information for the operator.

We present initial guidelines for designing interfaces for HRI, based on our observations in the study:

- Provide a map of where the robot has been. As seen in Section 5.3, operators using systems with maps were more successful in navigating arena area. Without a map, the operator must try to track the robot's path in his head.

- Provide fused sensor information to lower the cognitive load on user. In the three interfaces with multiple data types (Systems A, B, and C), all required the user to mentally fuse video with other sensor streams.
- Provide UIs that support multiple robots in a single display. We saw in Section 5.3 that the number of commands doubled when two robots were used instead of one. These commands needed to be entered in two separate windows.
- Minimize the use of multiple windows. With additional sensor fusion, more information could be displayed in a single window.
- Provide more spatial information about the robot in the environment. Spatial information could take the form of a map, discussed earlier, or some other method. At the very least, operators must be aware of their robots' immediate surroundings to avoid bumping into obstacles or victims.
- Provide robot help in deciding which level of autonomy is most useful. Team B's system had four levels of autonomy available, and the operator needed to select the method appropriate for this situation. The sensor data on the robot could be processed to assist with this decision. For example, we noticed that Team B's operator changed to autonomous mode whenever he felt that he was in a very tight situation; the robot could easily automate this switch or the suggestion of this switch.

This article contains evaluation guidelines and coding methods that may be used as frameworks for organizing results of future evaluations. We encourage other researchers in the HRI field to utilize and extend these frameworks to maximize our ability to learn from future studies and to be able to quickly transfer results into practice.

NOTES

Acknowledgments. Many people assisted with the study, including Brian Antonishek (NIST), Mike Baker (UMass Lowell), Jenn Casper (American Standard Robotics), Adam Jacoff (NIST), Elena Messina (NIST), Mark Micire (American Standard Robotics), Jesse Miyoshi (NIST), Justin Osborn (NIST), Craig Schlenoff (NIST), Phil Thoren (UMass Lowell), Brian Weiss (NIST), the participating teams, and the domain expert. Thanks to the editors, the reviewers and Marian Williams for their suggestions to improve this article.

Support. This work was supported in part by NSF IIS-0308186 and the DARPA MARS Program.

Authors' Present Addresses. Holly A. Yanco, Computer Science Department, University of Massachusetts Lowell, One University Avenue, Olsen Hall, Lowell, MA 01854. E-mail: holly@cs.uml.edu. Jill L. Drury, The MITRE Corporation, Mail Stop K320, 202 Burlington Road, Bedford, MA 01730. E-mail: jldrury@mitre.org. Jean Scholtz, NIST, 100 Bureau Drive, MS 8940, Gaithersburg, MD 20899. E-mail: jean.scholtz@nist.gov.

HCI Editorial Record. First manuscript received December 9, 2002. Revision received July 7, 2003. Accepted by Sara Kiesler and Pamela Hinds. Final manuscript received September 15, 2003. — *Editor*

REFERENCES

AAAI/RoboCup Rules Committee. (2002). *AAAI-2002 Robot Rescue Competition rules*. Retrieved December 1, 2002, from http://www.rescue-robotics.com/RescueRules/RobotRescue2003/2002Rules/rules.html

Burke, J. L., Murphy, R. R., Coovert, M. D., & Riddle, D. L. (2004). Moonlight in Miami: A field study of human–robot interaction in the context of an urban search and rescue disaster response training exercise. *Human–Computer Interaction, 19,* 85–116. [this special issue]

Casper, J. (2002). *Human–robot interactions during the robot-assisted urban search and rescue response at the World Trade Center.* Unpublished master's thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, FL.

Draper, J. V., Pin, F. G., Rowe, J. C., & Jansen, J. F. (1999). Next generation munitions handler: Human–machine interface and preliminary performance evaluation. *Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems*. New York: ACM.

Drury, J. L., Scholtz, J., & Yanco, H. A. (2003). Awareness in human–robot interactions. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*. Washington, DC: IEEE.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review, 87,* 215–251.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Fong, T. W. (2001). *Collaborative control: A robot-centric model for vehicle teleoperation*. Technical Report CMU–RI–TR–01–34. Pittsburgh, PA: Carnegie Mellon University Robotics Institute.

Jacoff, A., Messina, E., & Evans, J. (2000). A standard test course for urban search and rescue robots. *Proceedings of the Performance Metrics for Intelligent Systems Workshop*. Gaithersburg, MD: National Institute of Standards and Technology.

Jacoff, A., Messina, E., & Evans, J. (2001). A reference test course for autonomous mobile robots. *Proceedings of the SPIE-AeroSense Conference.* Bellingham, WA: International Society for Optical Engineering.

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *The handbook of human–computer interaction* (pp. 135–157). Amsterdam: North-Holland.

Leveson, N. G. (1986). Software safety: Why, what and how. *Computing Surveys, 18,* 125–162.

Messina, E., Meystel, A., & Reeker, L. (2001). Measuring performance and intelligence of intelligent systems: PerMIS 2001 white paper. *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop*. Mexico City: IEEE.

Nielsen, J. (1993). *Usability engineering*. Chestnut Hill, MA: AP Professional.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Proceedings of the CHI 94 Conference on Human Factors in Computing Systems*. Boston, MA: ACM.

Scholtz, J. (2002). Evaluation methods for human–system performance of intelligent systems. *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*. Gaithersburg, MD: National Institute of Standards and Technology.

Simmons, R., Goldberg, D., Goode, A., Montemerlo, M., Roy, N., Sellner, B., et al. (2003). GRACE: An autonomous robot for the AAAI Robot Challenge. *AI Magazine, 24,* 51–72.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F. et al. (2000). Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research, 19,* 978–999.

Yanco, H. A. (2000). *Shared user–computer control of a robotic wheelchair system*. Unpublished doctoral dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Yanco, H. A. (2001). Designing metrics for comparing the performance of robotic systems in robot competitions. *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop*. Mexico City: IEEE.

Yanco, H. A., & Drury, J. L. (2002). A taxonomy for human–robot interaction. *Proceedings of the AAAI 2002 Fall Symposium on Human–Robot Interaction* (Technical Report FS–02–03). Falmouth, MA: AAAI.